

PROGRAMMER TO PROGRAMMER™



Professional ASP.NET 2.0 XML

ASP.NET 2.0 XML

高级编程

(第3版)

(美) Thiru Thangarathinam 著

肖奕 曹俊 译



清华大学出版社

ASP.NET 2.0 XML

高级编程

(第 3 版)

(美) Thiru Thangarathinam

著

肖 奕 曹 俊

译

清华大学出版社

北 京

Thiru Thangarathinam

Professional ASP.NET 2.0 XML

EISBN: 0-7645-9677-2

Copyright©2006 by Wiley Publishing, Inc.

All Rights Reserved. Authorized translation from the English language edition published by Wiley Publishing, Inc.

本书中文简体字版由 Wiley Publishing, Inc.授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2006-1584

版权所有, 翻印必究。举报电话: 010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

本书防伪标签采用特殊防伪技术, 用户可通过在图案表面涂抹清水, 图案消失, 水干后图案复现; 或将表面膜揭下, 放在白纸上用彩笔涂抹, 图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

ASP.NET 2.0 XML 高级编程(第3版)/(美) 然哥诺辛尔蒙(Thangarathinam,T.)著; 肖奕 曹俊 译.

—北京: 清华大学出版社, 2006.9

书名原文: Professional ASP.NET 2.0 XML

ISBN 7-302-13214-3

I.A… II.①然…②肖…③曹… III. ①主页制作—程序设计 ②可扩充语言, XML—程序设计 IV. ①TP393.092 ②TP312

中国版本图书馆 CIP 数据核字(2006)第 064853 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

组稿编辑: 曹 康 文稿编辑: 李 阳

封面设计: 康 博 版式设计: 康 博

印 刷 者: 清华大学印刷厂

装 订 者: 三河市金元印装有限公司

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 33.5 字数: 815 千字

版 次: 2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷

书 号: ISBN 7-302-13214-3/TP·8353

印 数: 1~4000

定 价: 59.90 元

前 言

本书将交替讲述两种强大的技术：ASP.NET 和 XML。

XML 早已成为热点。业内对这种允许以独立于平台的方式进行数据通信和信息存储的 W3C 建议标准的广泛认同令人吃惊。您在任何地方都可以看到 XML 的使用——从使用转换语言 XSLT 在不同的浏览器上显示数据，到使用 SOAP 在 Web 服务之间传递消息。

.NET 是 Microsoft 的革新和足以夸耀的新概念。.NET 具有一些功能，例如允许以独立于语言的方式开发应用程序，在语言之间共享代码，自描述类以及自文档化 (self-documenting) 程序代码命名。.NET (特别是 ASP.NET) 就是专门为 Web 服务和便捷开发而设计的。随着 .NET 2.0 Framework 的发布，.NET 包括了对 ASP.NET 所有领域的明显改善。对 Web 页面开发来说，新的 XML 数据控件，例如 XmlDataSource 和 TreeView，使得无需编写代码即可在 ASP.NET Web 页面上显示和编辑数据，这样在某些情况下可以减少 70% 的所需代码量。ADO.NET 2.0 包括了很多新的特性，让您体验由 SQL Server 2005 (SQL Server 的下一个主要版本) 带来的 XML 新特性。

为了能够获得这种令人兴奋的新的 Web 编程环境，Microsoft 最大限度地利用了 XML。实际上，没有一种技术能像 XML 一样与 ASP.NET 结合得如此紧密。它可以作为任何数据的通用数据格式，从配置文件到元数据、Web Service 通信以及对象串行化。在 System.Xml 命名空间中的所有 XML 功能都被显著增强，比如提高性能和对标准化的支持。一些与 XML 相关的主要技术改善有：处理内存中 XML 数据的新模式、可编辑的 XPathNavigator、新的 XSLT 处理器以及对 XmlReader 和 XmlWriter 类的强类型支持等。与此相关联的是 ADO.NET 2.0 中对 XML 的新支持。因为有了 ADO.NET 2.0 的新特性，编程人员现在就可以同时使用分层 XML 和关系型数据库格式来访问和更新数据了。

0.1 本书读者对象

本书适合于已经开始进行 ASP.NET 开发和已经熟悉 XML 的中级或高级编程人员。当我在第 2 章介绍 ASP.NET 2.0 的很多新概念的时候，我并不想让本书成为开发人员学习 ASP.NET 的第一个切入点，因为在这个方面已经有大量的书籍和文章。而我将单刀直入地讲解如何在 ASP.NET Web 应用程序中使用 XML。为了能让本书产生最大效用，您最好具备一些基本的 C# 知识。本书所有的代码示例都用 C# 编写。

同样地，在使用本书的同时您可能还需要很多其他介绍 XML 技术的书籍和文章。我认为是 XML、命名空间和 XSLT 的基础知识以及对 XML 模式的基本了解。

0.2 本书主要内容

本书讲述 XML 的一系列特性以及在 ASP.NET 中如何应用它们开发 Web 应用程序。XML 在 .NET Framework 中无处不在,从串行化到 Web 服务,从数据访问到配置。在本书的第一部分中,将会深入介绍在 .NET 平台中实现 XML 的关键类。该部分将通过 ASP.NET 示例和参考信息来讨论读写器、验证、模式以及 XML DOM。接着本书将讲解 XPath 和 XSL 转换(XSLT)、ADO.NET 中对 XML 的支持以及如何使用 XML 来显示数据。

本书最后部分的重点在 SQL Server 2005 的 XML 特性、XML 串行化、XML Web 服务以及涉及到基于 XML 的配置文件和文件的 XML 扩展。本书还介绍了一些与 ASP.NET 和 Web 服务相关的 XML 案例分析,让您真实地体验这些特性。

0.3 本书结构

本书共包含 15 章,还包括两个案例分析。本书将让读者学习在 ASP.NET 2.0 中进行 XML 开发的整个过程。将采取重点讲述的方式,只为读者讲解每个阶段所必需的知识,而不使用附加材料、过于复杂的技术术语或者毫无用处的章节来详细讨论一些规范和标准。下面简单介绍一下这些章节。

0.3.1 XML 介绍

XML 出现在很多商业应用中并不断出现在日常生活中,它为那些打算使用 Web 服务来交换文档的公司提供了一种通用的数据格式。本章讲述作为一种语言的 XML 及其相关技术。本章中将着重介绍的 XML 技术有:XML 文档元素、命名空间、实体、DTD、XDR、XSD、XSD 模式数据类型、XSLT、XML DOM、XPath、SAX、XLink、XPointer 和 XQuery。

0.3.2 ASP.NET 2.0 介绍

第 2 章将简要介绍 ASP.NET 2.0 的新特性。将重点讲解新的 ASP.NET 页面体系结构、新的数据控件和代码共享特性。书中会提问“什么是主控页面(master page)”,然后讲解如何通过主控页面和主题来创建统一的 Web 站点。接着会讲述安全控件和 Web parts 架构并演示 ASP.NET 2.0 是如何能够减少 70% 的代码量的。最后,将讲述 ASP.NET 2.0 的新缓存和管理以及控制功能。

0.3.3 .NET Framework 中的 XML 类

第 3 章简要介绍 .NET Framework 中所有新的 XML 类,这些内容也会在本书的其他章节中详细讨论。

Microsoft 在 .NET 2.0 中引入了很多 XML 的新应用,也为改善 XML API 内核做了一些革新工作。本章将从在配置文件中开始使用 XML 开始,讨论 DOM、XSD 模式验证、XSLT 转

换、XML 串行化、Web 服务和 ADO.NET 中对 XML 的支持，以及这些内容所涉及的命名空间和类。本章还将演示新的 ASP.NET 配置改善并快速浏览一下 .NET Framework 2.0 中的配置类。

0.3.4 读写 XML 数据

从第 4 章开始的几个章节(第 4~6 章)将详细讨论包含在 System.Xml 内的功能。

本章将特别介绍 .NET Framework 提供的、用于读写 XML 文档的快速且只向前的只读机制，即 XmlReader 和 XmlWriter 类。书中将剖析新的 XML 读写模式并介绍多种使用方式，供您用来读写 XML 数据。还将讨论节点顺序、解析属性、定制读写器设置、空格处理和命名空间处理等，以及其他一些命名空间的支持。

0.3.5 验证 XML 数据

第 5 章将介绍不同的 XML 验证语法：DTD、XDR 模式和 XSD 模式。还会介绍可用于在 Visual Studio 2005 中创建 XSD 模式的所有方法：从 DTD 使用 XML 设计器、从 XML 文档、XDR 模式或程序集使用 XSD 生成器。本章还会讨论模式对象并介绍如何将 XML 文档链接至 DTD、XDR 模式和 XSD 模式，并且介绍如何使用 XmlReaderSettings 联合 XmlReader 类来执行验证。本章还将演示使用 XmlSchemaSet 类来保持内存中模式的缓存、优化性能并介绍如何处理 XML 文档中无限定/命名空间限定的内容。

0.3.6 XML DOM 对象模式

第 6 章将介绍类的 System.Xml 命名空间中提供的 .NET Framework 内的 DOM 功能。本章将讲解用程序来创建 XML 文档、通过 URL 或者内存中的字符串来打开文档以及在将文档串行化成 XML 字符串之前检索并访问文档内容。本章还会介绍 XmlDocument 对象和 XmlReader、XmlWriter 对象之间的区别以及何时使用其中之一最为恰当。最后将演示 XmlDocument 类的 XPath 功能，并重点讲解 XPathNavigator 类用于修改内存中 XML 文档的新的编辑功能。

0.3.7 使用 XSLT 转换 XML 数据

.NET Framework 为处理 XSLT 和 XPath 提供了强有力的支持，在 .NET Framework 2.0 中，XSL 已经完全被改写并引入了全新的 XSLT 处理器。第 7 章将介绍用于在 .NET Framework 中进行 XSL 转换的技术，即 System.Xml.Xsl 命名空间和 System.Xml.XPath 命名空间，还有最新引入的 XslCompiledTransform 类。.NET Framework 完全支持由 W3C 定义的 XSLT 和 XPath 规范，而且还提供很多对这些规范有用的扩展来加强在 .NET 应用程序中样式表的使用。接着介绍了使用 <msxsl:script> 嵌入脚本来转换 XML 文档并演示如何用扩展对象来扩充样式表。在本章的结尾将讨论一些高级的 XSLT 操作，例如如何将一个节点集传递给样式表和如何使用 XmlResolver 来解析外部样式表。

0.3.8 ADO.NET 中对 XML 的支持

第 8 章将开始介绍类的 `System.Xml` 命名空间, 探索在 .NET 中特别是 ADO.NET 中如何使用 XML 的广阔领域以及数据访问技术。

第 8 章将介绍在 ADO.NET 2.0 中 XML 所扮演的角色并重点讲解与 ADO.NET 相关的 XML 特性。本章将介绍 `DataSet` 和 `DataTable` 类的功能, 包括读写 XML 和利用程序访问或更改它的 XML 表示形式。还将重点介绍如何用 `XmlDataDocument` 来同步 `DataSet` 以及为什么需要这么做。本章还会介绍强类型 `DataSet` 的创建及其优点。本章的最后会粗略介绍如何从 ADO.NET 访问 SQL Server 2005 中可用的 XML 新特性。

0.3.9 XML 数据显示

ASP.NET 中支持的 XML 提供了对存储、获取和生成 XML 的最佳支持。本章将介绍 `web.sitemap` 文件, 通过这个文件可以保存 Web 站点的层次结构并驱动 Web 站点的导航结构。随后, 将讨论 XML 数据控件的新特性, 例如用 `XmlDataSource`、`TreeView` 和 `GridView` 在浏览器上直接使用并显示本地的 XML。本章的最后将介绍新的 ASP.NET 2.0 脚本回调功能, 该功能用来直接从浏览器获取 XML 数据而无需刷新页面。

0.3.10 SQL Server 2005 的 XML 整合

随着 SQL Server 2005 的发布, 对 XML 的支持达到了一个高度, SQL Server 2005 提供了对 XML 数据强大的 XML 查询和数据修改功能。从本章开始将介绍 SQL Server 2005 中的 XML 新特性, 包括 FOR XML 子句的增强、XQuery 的支持以及 XML 数据类型。然后将讨论在 ADO.NET 中同步和异步对 FOR XML 查询语句的执行。本章还将讨论处理类型化和非类型化 XML 数据类型列的步骤。最后将演示如何使用 ADO.NET 从类型化列中获取 XSD 模式并重点讲解 `MARS` 和 `OPENXML()` 函数。

0.3.11 使用 ASP.NET 2.0 和 SQL Server 2005 创建飞机订票系统

本实例将到目前为止本书所介绍的所有概念联系起来进行研究, 包括 XML DOM、ADO.NET 中 XML 的支持、.NET 中 XSLT 特性和 XML 数据显示。本案例分析的重点在于现实的飞机订票 Web 站点中这些 XML 特性的协同工作以及展示使用这些特性的最佳实践。本章还将讨论 N 层设计方法并演示如何利用这个方法创建一个可扩展的灵活的飞机订票系统。

0.3.12 XML 串行化

在第 12 章中, 将介绍使用 `XmlSerializer` 类从 `System.Xml.Serialization` 命名空间将 XML 文档串行化为 XML 数据。更一般的情况是, 创建串行器, 然后将普通类型、复杂对象、属性、枚举值、数组和复合对象进行串行化和反串行化。本章还将介绍使用嵌套对象进行

串行化和反串行化，接着介绍格式化 XML 文档、XML 属性和文本内容。在本章的结尾，还将讨论使用新的 XML 串行器生成器工具通过预生成的程序集来提高串行化性能。

0.3.13 XML Web Services

Web Services 是可以从任何客户端通过 HTTP 调用的一套对象和方法。Web Services 建立在简单对象访问协议(Simple Object Access Protocol, SOAP)上。本章将通过使用 .NET Framework 2.0 和 Visual Studio 2005 创建 XML Web Services 来让您了解 XML Web Services。在开始的讨论之后，将接着讨论高级的 Web Services 概念，例如 SOAP 头文件、SOAP 扩展、XML 串行化定制、模式导入扩展、异步 Web Services 方法以及 Web Services 方法的异步调用等。

0.3.14 ASP.NET 2.0 配置

本章将介绍 ASP.NET 2.0 的新的配置管理 API，可以让用户通过建立程序或脚本来创建、读取和更新 web.config 和 machine.config 文件中的设置。本章还将讨论内置于 IIS Administration MMC 中新全面的的管理工具，能让管理员通过图形化界面来读取或者更改 XML 配置文件中的任何设置。在整章中，将重点讲述新的配置管理类、属性和配置 API 的方法，还将提供如何在 ASP.NET 应用程序中使用它们的示例。

0.3.15 使用 XML Web Services 建立 ShoppingAssistant

本章的讲述基于一个名为 ShoppingAssistant 的案例分析，该案例为顾客提供了一站式的服务，让他们能够找到所需的信息，例如在售商品、不同商场可买到的商品，以及不同商场中相同商品的价格比较等。在这个案例分析中，将演示如何在真实的 Web 应用程序中使用异步 Web 服务调用功能并结合其他 .NET 特性(如 XML Serialization、FileSystemWatcher 和 Timer 组件)来体验 Web 服务。

0.4 使用本书的条件

本书中所有的示例都是 ASP.NET 示例。所以运行这些应用程序主要需要 .NET Framework 2.0 和 Microsoft Visual Studio 2005。您还需要有 SQL Server 2005 服务器并安装了 AdventureWorks 示例数据库以便能够运行大多数示例。有些示例将使用 SQL Server 2005 的 Express 数据库。

本书中的 SQL Server 示例利用了集成的安全来连接至 SQL Server 数据库，所以请记住启用 SQL Server 中的集成身份验证。这还需要打开 ASP.NET Web 站点中的集成 Windows 身份验证(以及根据您配置的角色扮演)。

0.5 源代码

在读者学习本书中的示例时，可以手工输入所有的代码，也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书 Web 站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/download 上下载。登录到站点 <http://www.wrox.com/>，使用 Search 工具或使用书名列表就可以找到本书。接着单击本书细目页面上的 Download Code 链接，就可以获得所有的源代码。

注释：

由于许多图书的标题都很类似，所以按 ISBN 搜索是最简单的，本书英文版的 ISBN 是 0-7645-9677-2。

在下载了代码后，只需用自己喜欢的解压缩软件对它进行解压缩即可。另外，也可以进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页，查看本书和其他 Wrox 图书的所有代码。

0.6 勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误，但是错误总是难免的，如果您在本书中找到了错误，例如拼写错误或代码错误，请告诉我们，我们将非常感激。通过勘误表，可以让其他读者避免受挫，当然，这还有助于提供更高质量的信息。

请给 fwkbook@tup.tsinghua.edu.cn 发电子邮件，我们就会检查您的信息，如果是正确的，就把它传送到该书的勘误表页面上，或在本书的后续版本中采用。

要在网站上找到本书的勘误表，可以登录 <http://www.wrox.com>，通过 Search 工具或书名列表查找本书，然后在本书的细目页面上，单击 Book Errata 链接。在这个页面上可以看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表，网址是 www.wrox.com/misc-pages/booklist.shtml。

0.7 P2P.WROX.COM

要与作者和同行讨论，请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统，便于您张贴与 Wrox 图书相关的消息和相关技术，与其他读者和技术用户交流心得。该论坛提供了订阅功能，当论坛上有新的消息时，它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上，有许多不同的论坛，它们不仅有助于阅读本书，还有助于开发自己的应用程序。要加入论坛，可以遵循下面的步骤：

- (1) 进入 p2p.wrox.com，单击 Register 链接。
- (2) 阅读使用协议，并单击 Agree。

- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息，单击 **Submit**。
- (4) 您会收到一封电子邮件，其中的信息描述了如何验证账户，完成加入过程。

注释：

不加入 P2P 也可以阅读论坛上的消息，但要张贴自己的消息，就必须加入该论坛。

加入论坛后，就可以张贴新消息，响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要让该网站给自己发送特定论坛中的消息，可以单击论坛列表中该论坛名旁边的 **Subscribe to this Forum** 图标。

关于使用 Wrox P2P 的更多信息，可阅读 P2P FAQ，了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ，可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 XML 介绍.....1	3.1.5 XPath 支持.....44
1.1 XML 入门.....1	3.1.6 XML 模式对象模型.....45
1.1.1 自描述数据.....2	3.1.7 理解 XML 验证.....46
1.1.2 基本术语.....3	3.1.8 用 XSLT 转换 XML 数据.....46
1.1.3 XML 文档的组成部分.....4	3.1.9 XML 串行化.....47
1.2 XML 技术.....11	3.1.10 XML Web 服务.....49
1.2.1 DTD.....12	3.1.11 XML 和 ADO.NET.....52
1.2.2 XDR.....13	3.1.12 ASP.NET 配置.....53
1.2.3 XSD.....13	3.2 小结.....55
1.2.4 XSLT.....17	第 4 章 用 XmlReader 和 XmlWriter
1.2.5 XML DOM.....17	读写 XML 数据.....57
1.2.6 XPath.....18	4.1 XML 的读取器和书写器.....57
1.2.7 SAX.....19	4.2 使用 XmlReader 类读取 XML.....59
1.2.8 XLink 和 XPointer.....19	4.2.1 XmlReader 简介.....59
1.2.9 XQuery.....19	4.2.2 使用 XmlReader 类读取 XML 数据的步骤.....59
1.3 XML 的优势.....20	4.3 写入 XML 数据.....77
1.4 小结.....20	4.4 小结.....90
第 2 章 ASP.NET 2.0 介绍.....21	第 5 章 XML 数据验证.....93
2.1 ASP.NET 2.0 的功能.....21	5.1 XML 验证.....93
2.1.1 开发人员的效率.....21	5.1.1 .NET Framework 2.0 支持的验证类型.....94
2.1.2 控制和管理.....33	5.1.2 使用 XSD 模式验证 XML 数据.....95
2.1.3 速度和性能.....35	5.1.3 模式的缓存.....100
2.2 小结.....37	5.1.4 XML DOM 验证.....103
第 3 章 .NET Framework 中的 XML 类.....39	5.1.5 使用内嵌模式验证 XML.....105
3.1 .NET Framework 2.0 对 XML 的支持.....39	5.1.6 使用 DTD.....108
3.1.1 .NET Framework 2.0 对 XML 支持的设计目的.....39	5.1.7 使用 Visual Studio 2005 创建 XML 模式.....113
3.1.2 XML 命名空间.....40	5.1.8 .NET 模式对象模型(SOM).....116
3.1.3 XML 解析.....41	5.1.9 通过程序从 XML 文件推断 XSD 模式.....123
3.1.4 编写 XML.....43	

5.2 小结	124
第6章 XML DOM 对象模型	125
6.1 探索 DOM 处理	125
6.2 用 XML 文档对象模型编程	127
6.2.1 文档类	128
6.2.2 集合类	129
6.2.3 XmlDocument 类	129
6.2.4 使用 XmlDocument 类	132
6.2.5 通过程序创建 XML 文档	141
6.2.6 XmlDocumentFragment 类	151
6.2.7 XML DOM 对 XPath 的支持	152
6.2.8 在 XmlDocument 中验证 XML	163
6.3 小结	163
第7章 用 XSLT 转换 XML 数据	165
7.1 XSLT 入门	165
7.1.1 XSLT、XSL 和 XPath 概述	166
7.1.2 XSLT 的优势	167
7.1.3 XSLT 元素	168
7.1.4 XSLT 函数	170
7.1.5 将 XSL 样式表应用于 XML 文档	171
7.1.6 XSL 转换中涉及的 .NET 类	178
7.1.7 XSL 样式表中用户自定义函数	184
7.1.8 XsltSettings 类	189
7.1.9 一个完整的示例	190
7.1.10 高级 XSLT 操作	198
7.1.11 调试 XSLT 样式表	201
7.2 小结	203
第8章 XML 和 ADO.NET	205
8.1 ADO.NET 和 XML	205
8.1.1 载入 XML 至 DataSet	206
8.1.2 DataSet 模式	210
8.1.3 转换 DataSet 为 XML	214
8.1.4 有类型 DataSet	222
8.1.5 XmlDocument 对象和 DataSet	227
8.1.6 XmlDocument 和 XPathNavigator 之间的关系	233
8.1.7 DataTable 和 XML	235
8.2 小结	237
第9章 显示 XML 数据	239
9.1 ASP.NET 2.0 层次数据控件	239
9.1.1 站点导航	240
9.1.2 XmlDataSource 控件	243
9.1.3 缓存	254
9.1.4 XML Web 服务器控件	257
9.2 客户端 XML	264
9.2.1 ASP.NET 2.0 回调功能	264
9.2.2 ASP.NET 的 Atlas 技术	271
9.3 小结	276
第10章 SQL Server 2005 的 XML 集成	277
10.1 SQL Server 2005 中新 XML 特性	278
10.1.1 SQL Server 2005 中的 FOR XML	278
10.1.2 从 ADO.NET 执行 FOR XML 查询	280
10.1.3 SQL Server 2005 中的 XML 数据类型	287
10.1.4 处理 ADO.NET 的 XML 数据类型列	292
10.1.5 在客户端使用 XML 模式	305
10.1.6 ADO.NET 中的多活动结果集	311
10.1.7 XML 数据类型和 DataSet	315
10.1.8 OPENXML()	318
10.1.9 其他 XML 特性	320
10.2 小结	322

第 11 章 使用 ASP.NET 2.0 和 SQL Server 2005 构建飞机订票系统	323
11.1 案例分析概述	323
11.1.1 系统的体系结构	323
11.1.2 业务流程	324
11.2 实现	325
11.2.1 数据库设计	325
11.2.2 AirlineReservationsLib 组件的实现	329
11.2.3 Web 站点的实现	337
11.2.4 整合所有内容	362
11.3 小结	363
第 12 章 XML 串行化	365
12.1 串行化入门	366
12.1.1 XmlSerializer 类	366
12.1.2 高级串行化	372
12.1.3 反串行化 XML	382
12.1.4 泛型和 XML 串行化	390
12.1.5 预生成串行化程序集	394
12.1.6 处理异常	395
12.2 小结	396
第 13 章 XML Web 服务	399
13.1 XML Web 服务	399
13.1.1 构建 ASP.NET Web 服务	400
13.1.2 为 Web 服务创建代理类	404
13.1.3 返回复杂类型	407
13.1.4 使用 SOAP 头	417
13.1.5 使用 SOAP 扩展	422
13.1.6 来自客户机应用程序的 Web 服务异步调用	429
13.1.7 用 IE Web 服务行为从浏览器进行 Web 服务的异步调用	433
13.1.8 异步 Web 服务方法	440
13.1.9 用 IXmlSerializable 控制 XML 串行化	442
13.1.10 使用 Schema Importer Extension	445
13.1.11 .NET Framework 2.0 中的其他 Web 服务功能	448
13.2 小结	449
第 14 章 ASP.NET 2.0 配置	451
14.1 ASP.NET 配置	451
14.1.1 配置层次结构	452
14.1.2 访问配置节的 ASP.NET 1.x 方式	453
14.2 ASP.NET 2.0 配置管理	453
14.2.1 ASP.NET 2.0 中的新配置节	453
14.2.2 WebConfigurationManager 类	456
14.2.3 从预定义节中检索配置	457
14.2.4 加密和解密配置节	462
14.2.5 枚举配置节	465
14.2.6 读配置节	467
14.2.7 创建自定义配置节	471
14.2.8 内置配置管理工具	475
14.3 小结	478
第 15 章 用 XML Web 服务构建 ShoppingAssistant	479
15.1 ShoppingAssistant 案例分析	479
15.1.1 ShoppingAssistant 的体系结构	480
15.1.2 业务过程	481
15.2 实现	482
15.2.1 数据库设计	482
15.2.2 ContentPublisher Web 服务的实现	484
15.2.3 ShoppingAssistant 组件的实现	492
15.2.4 ShoppingAssistant Web 应用程序的实现	494
15.2.5 使用 Web 服务和 Windows 服务的异步调用	506

15.2.6 修改 ShoppingAssistant Web 页面, 以使用 XML 文件	511	15.2.7 实现简化报表数据收集 的 FileSystemWatcher	513
		15.2.8 综合运用以上内容	518
		15.3 小结	518

第 1 章

XML 介绍

可扩展标记语言(Extensible Markup Language, XML)是由 World Wide Web Consortium(W3C, <http://www.w3c.org>)定义的一种语言,是为 Web 设定标准的主体。可以使用 XML 创建自己的元素,进而创建自己使用的自定义标记语言。通过这种方式,XML 取代了其他标记语言,例如超文本标记语言(Hypertext Markup Language, HTML);在 HTML 中,使用的所有元素都是预先定义好的——这远不够用。实际上,XML 是一种元标记语言,因为它让您创建自己的标记语言。

XML 是完全开发 Internet 和 Web 潜力的理想方式。就像 HTML、超文本传输协议(HyperText Transfer Protocol, HTTP)和 Web 浏览器为网络计算机和大众之间的通信方式铺设了令人兴奋的大道一样,XML 及其相关技术开辟了大众和计算机之间电子化通信的新领域。但是,对于作为“最小公分母”语言并可用于其他所有(私有的或开放的)系统的 XML 来说,它的承诺是能够同时实现人-机和机-机通信。

XML 来自于与 Web 的强力结合。Web 提供了一系列传输数据的协议;而 XML 提供了定义这些数据的方式。我们来看看企业采取了新的方式而产生的最直接的效果。替换了联系紧密的服务器网络之后,现在企业通过 Web 的全球范围和领域围绕的并不是传统上的网络而是 Web 本身。XML 已经毫无疑问地成为了通用数据共享的标准。随着 XML 应用的不断增长,XML 实现的方式也在不断增加。XML 可用于各种用途,从显而易见的工作,如标记简单数据文件和存储临时数据,到复杂的工作,例如在两个程序或进程之间传递信息。

XML 出现在很多商业应用中并不断出现在日常生活中,它为那些打算交换文档的公司提供了一种通用的数据格式。Web 服务以独立于平台的方式利用它对消息和数据进行编码。它甚至还作为将内容和表现完全分离的工具而被用来构建 Web 站点。

本章介绍作为语言的 XML 及其相关技术。完整地讨论这个主题将会占用大量的篇幅,所以本章努力在详细和简明之间找到折衷。在本章后面,您将学习不同的 XML 相关技术和它们的用法。但是,在这之前,先简单地介绍 XML 本身。

1.1 XML 入门

XML 来自于标准通用标记语言(Standard Generalized Markup Language, SGML),一种主要用于大型文件项目、具有强大功能的语言。XML 的设计者从 SGML 中吸收了大量元

素并从 HTML 中汲取了经验。他们虽然制定出了一个只有 SGML 规范 20%大小的规范,却相当完善。虽然 SGML 一般只适用于需要工业级语言的用户,但是 XML 则适用于任何人。

XML 的一个强大之处在于它为表带来了可扩展性。XML 没有任何自己的标签,也不限制您喜欢其他的标记语言。相反,XML 为开发您自己的语义标签定义了规则。您自己创建的标签将形成词汇表,可用于将数据放入信息的结构树中。您可以将 XML 看成是一种元标记语言,可以让开发人员、公司甚至行业来创建他们自己的特定标记语言。

在 XML 中需要掌握的最重要的概念之一是内容,而不是表现。您创建的标签主要用于组织数据而不是用于显示数据。比如,XML 不用于在一个新的段落中标明文档中特殊的部分,或另外一个部分需要粗体字显示。XML 用于开发标签来标明一段特定的数据是作者的姓,另外一段是书的标题,而第 3 段是书的出版年份。

1.1.1 自描述数据

正如前面所提到的,XML 最强大的特性就是它并没有定义任何标签。创建自己标签的目的是让 XML 充满灵活性;但是,要定义出有意义的标签就取决于您自己了。在创建标签时,没有必要简写或缩短标签名称。这并不会使得处理它们的速度有所加快,但是却会让 XML 文档更加混乱或者难以理解。请记住,开发人员将会按照 XML 文档来编写代码。一方面,您当然可以按如下方式定义标签:

```
<H1>XSLT Programmers Reference
<p><b>Michael Kay</b></p>
</H1>
```

使用这些基于 HTML 的标签可以使得浏览器中的显示很容易,但是它们不会在文档中添加任何信息。记住,XML 关注的是内容而不是表现。创建如下的 XML 会更加有意义:

```
<books>
  <book>
    <title>XSLT Programmers Reference</title>
    <author>Michael Kay</author>
  </book>
</books>
```

第 2 个示例使用人类语言使其更加具有可读性,它还为非人类提供了更多的功能性和多功能性。通过这套标签,应用程序可以很容易地访问书的标题或者作者姓名,而无需拆分任何字符串或者搜索空格。并且,对于开发人员编写代码来说,在 XML 文档中搜索作者的姓名将变得更加自然,例如当元素名称是 title 而不是 H1 的时候。

注意:

在前面示例中标签的缩进只是为了增加可读性,当然这并不是 XML 文档中所必需的。但是在创建文档时,您会发现缩进有助于您阅读。

为了处理前面的 XML 数据,不需要具体的编辑器来创建 XML 文档,虽然有不少的编

编辑器可供使用。并且也不需要尖端的技术。困扰于 XML 的恰恰是它的简单性。特别是，因为开发人员在设计中部署 XML 来简化工作的方式而让人们 XML 的兴趣大增。很多让软件开发人员年复一年重复做的艰巨工作现在都可以轻松完成了。XML 还让组件之间的通信更加容易，因为它提供了一种被当今大多数流行平台认可的标准化和结构化的语言。实际上，在 .NET 平台中，Microsoft 已经通过将 XML 作为整个平台的支撑来说明它的重要性。正如您将在后面章节中看到的，.NET 完全依赖于它架构中的 XML 和 SOAP(简单对象访问协议)以及基本服务来使得开发更加容易和高效。

1.1.2 基本术语

XML 术语大量地出现在 XML 社区中，有时多的让您无法接受。了解这些术语可以帮助您更多地了解关于 XML 的规范。

1. 格式良好(Well-Formed)

如果文档符合由 XML 规范定义的格式良好的规定，那么该文档就被认为是格式良好的。这些规定如下：

- 文档包含一个或多个元素。
- 文档只有一个根元素(也就是文档元素)。
- 元素的结束标签名称与开始标签的名称一致。
- 在一个元素中属性只能出现一次。
- 属性值不能包含有左尖括号(<)。
- 由开始和结束标签界定的元素必须互相正确地嵌套。

2. 有效性

首先，在考虑一个 XML 文档是否有效之前，必须考虑该文档是否是一个格式良好的文档。格式良好的要求应该相当简单，但关键是让 XML 文档从格式良好跨越到有效却有些困难。为了达到有效，XML 文档必须经过验证。可以通过文档类型定义(Document Type Definition, DTD)或者 XML 模式定义(XML Schema Definition, XSD)来对文档进行验证。对于将要被验证的 XML 文档来说，它必须符合相对应的 DTD 或者 XSD 模式表达的限制。

说明：

一个有效的文档并不能保证语义上完全正确。虽然 XML Schema 对元素和属性制定了比 XML DTD 更为严格的限制，但是它还是不能捕捉所有错误。例如，您可以定义一个需要两个小数位的价格数据类型，但是在您打算输入 16.00 时，却可以输入 1600.00，而模式文档却不能捕捉到这个错误。

在处理有效性的时候，必须记住 XML 文档可以有 3 种存在方式：

- 作为自由格式且格式良好的 XML 文档，没有 DTD 或者与其相关联的模式。
- 作为格式良好且有效的 XML 文档，符合 DTD 或者模式。
- 作为格式良好但无效的文档，因为它不符合由相关联的 DTD 或模式定义的限制。